Animation in React - A simple loading animation with Framer Motion - Animation in React doesn't have to be difficult

Written by Seth Corker on Benevolent Bytes

What we're making

A tweet showing the loading animation we created

This is a simple animation you've likely seen all over the internet and even sneaking into mobile apps too. It's a good indeterminate loading animation which signals to the user that something is happening and it's unclear how long it will take.

How to achieve it

Learn with the animation video tutorial

There are three main players at work to achieve this simple animation.

The core of our component looks like this.

```
<motion.div
  style={loadingContainer}
  variants={loadingContainerVariants}
  initial="start"
  animate="end"
>
  <motion.span
    style={loadingCircle}
    variants={loadingCircleVariants}
    transition={loadingCircleTransition}
  />
  <motion.span
    style={loadingCircle}
    variants={loadingCircleVariants}
    transition={loadingCircleTransition}
  />
  <motion.span
    style={loadingCircle}
    variants={loadingCircleVariants}
    transition={loadingCircleTransition}
  />
</motion.div>
```

Staggering Child Animations

Firstly, the container which holds our three circular spans has a variants prop, for both variants we specify.

```
const loadingContainerVariants = {
  start: {
    transition: {
      staggerChildren: 0.2,
    },
  },
  end: {
    transition: {
      staggerChildren: 0.2,
    },
  },
}
```

So for both our `start` and `end` variants, any animations which are triggered in the children (the moving circles) will be staggered. So the second circle will start playing its animation 200ms after the first and so on. Using this technique, it's very easy to fine-tune the animation to get the timing right and experiment to discover different animations.

Inheriting Variants

Secondly, the children, `<motion.span />` are using `loadingCircleVariants`. Here we specify the animation we want each loading circle to have (animating the vertical position). Because we have used the same name variants `start` and `end`, the children's animations will be triggered when the parent animates. The parent will ensure each child receives the event in a staggered fashion.

```
const loadingCircleVariants = {
  start: {
    y: "0%",
  },
  end: {
    y: "100%",
  },
}
```

The `loadingCircleVariants` object defining our variants

Never-ending Animation

Finally, we need to ensure the animation continues to play after it's been triggered.

```
const loadingCircleTransition = {
  duration: 0.5,
  yoyo: Infinity,
  ease: "easeInOut",
}
```

The `loadingCircleTransition` object defining the transition of our moving circles In Framer Motion, we use `yoyo` to have the animation reverse when it gets to the end and we give it the value of `Infinity` so it will play forever (we could specify a value if we wanted it to only repeat a number of times).

Where to go from here?

This animation is relatively easy to achieve in CSS and is probably already part of a library you're using but if you'd like to have more control over your animations, are learning about Framer Motion or want to leverage this animation library in more parts of your project - this is a simple way to start.

Source Code

You can see the source for the tutorial on the loading-animation repo on GitHub.